



PATENT ABSTRACTS OF JAPAN

(11) Publication number: **2001256072 A**(43) Date of publication of application: **21.09.01**

(51) Int. Cl.

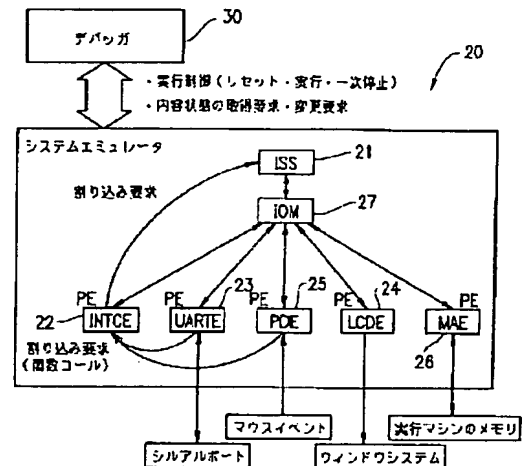
G06F 11/22**G06F 9/455**(21) Application number: **2000067805**(71) Applicant: **SHARP CORP**(22) Date of filing: **10.03.00**(72) Inventor: **KAMEI KATSUHIKO
TOMITA TSUNEO**(54) **SYSTEM EMULATOR**

(57) Abstract:

PROBLEM TO BE SOLVED: To improve the developing efficiency of a program by realizing an performance environment equivalent to a target machine on a performing machine.

SOLUTION: This system emulator 20 is constituted of an instruction set simulator ISS 21 for interpreting and performing an instruction set for a target machine and plural peripheral emulators PE22-26 for simulating each peripheral having different functions on the target machine. Each peripheral emulator PE22-26 simulates each peripheral based on the function of a peripheral to be simulated and a system corresponding to the complexity of the structure.

COPYRIGHT: (C)2001,JPO



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2001-256072
(P2001-256072A)

(43) 公開日 平成13年9月21日 (2001.9.21)

(51) Int.Cl. ⁷	識別記号	F I	テーマコード* (参考)
G 0 6 F 11/22	3 4 0	G 0 6 F 11/22	3 4 0 A 5 B 0 4 8
9/455		9/44	3 1 0 A

審査請求 未請求 請求項の数11 O L (全 21 頁)

(21) 出願番号 特願2000-67805 (P2000-67805)

(22) 出願日 平成12年3月10日 (2000.3.10)

(71) 出願人 000005049

シャープ株式会社

大阪府大阪市阿倍野区長池町22番22号

(72) 発明者 亀井 克比古

大阪府大阪市阿倍野区長池町22番22号 シ

ャープ株式会社内

(72) 発明者 富田 常雄

大阪府大阪市阿倍野区長池町22番22号 シ

ャープ株式会社内

(74) 代理人 100078282

弁理士 山本 秀策

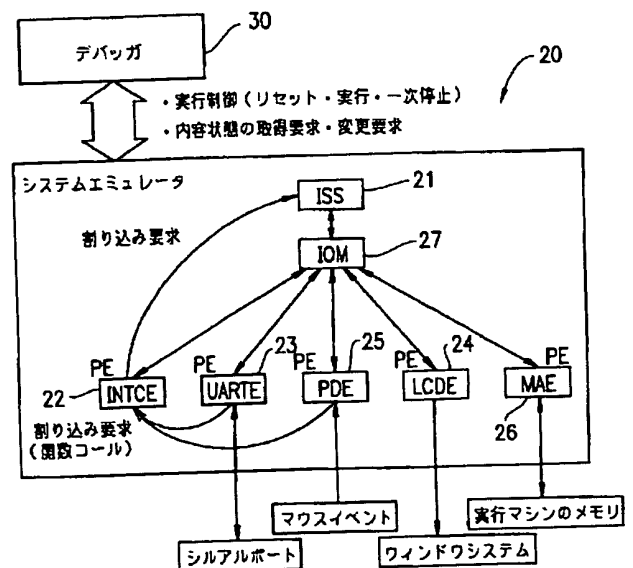
Fターム(参考) 5B048 BB02 DD14

(54) 【発明の名称】 システムエミュレータ

(57) 【要約】

【課題】 ターゲットマシンと同等の実行環境を実行マシン上で実現することによって、プログラムの開発効率を向上させる。

【解決手段】 ターゲットマシン用の命令セットを解釈、実行するインストラクションセットシミュレータISS 21と、ターゲットマシン上の異なる機能を有する各ペリフェラルをそれぞれ模倣する複数のペリフェラルエミュレータPE 22～26とによって、システムエミュレータ20が構成されている。各ペリフェラルエミュレータPE 22～26は、模倣するペリフェラルの機能、構造の複雑度に応じた方式に基づいて、各ペリフェラルの模倣をそれぞれ行う。



【特許請求の範囲】

【請求項 1】 実行マシン上にて、命令セットの異なるターゲットマシン用のプログラムを動作させるシステムエミュレータであって、

ターゲットマシンの命令セットを実行マシン上にて仮想的に実行するインストラクションセットシミュレータと、ターゲットマシン上にて異なる機能をそれぞれ実行する複数のペリフェラルの動作を、実行マシン上にてそれぞれ模倣するペリフェラルエミュレータとを具備し、各ペリフェラルエミュレータは、模倣するペリフェラルの機能、構造の複雑度に応じた方式によって、ターゲットマシン上での各ペリフェラルに対する操作を、実行マシン上での各ペリフェラルの操作にそれぞれ変換するようになっていることを特徴とするシステムエミュレータ。

【請求項 2】 前記各ペリフェラルエミュレータは、インストラクションセットシミュレータからペリフェラルのレジスタへのアクセスがあった場合に、そのアクセスの内容を解析し、対応する実行マシン上のペリフェラルのレジスタにアクセスを行うことによって、ターゲットマシン上でのペリフェラルへの操作を実行マシン上でのペリフェラルへの操作に変換する請求項 1 に記載のシステムエミュレータ。

【請求項 3】 前記ペリフェラルエミュレータは、インストラクションセットシミュレータからペリフェラルのレジスタへのアクセスがあった場合に、そのアクセスの内容を解析し、ペリフェラルエミュレータの各種設定などの内部状態を変化させるとともに、実行マシン上のペリフェラルにアクセスして、実行マシン上でのペリフェラルの状態を変化させて入出力させることによって、ターゲットマシン上でのペリフェラルへの操作を実行マシン上でのペリフェラルへの操作に変換する請求項 1 に記載のシステムエミュレータ。

【請求項 4】 前記ペリフェラルエミュレータは、ペリフェラルを扱う同一インターフェースのライブラリを、ターゲットマシン用のライブラリとは別に有しており、プログラムを実行する際に、プログラムからライブラリを呼び出し、ライブラリによってペリフェラルエミュレータの内部状態を変化させ、実行マシン上のペリフェラルにアクセスし、実行マシン上でのペリフェラルの状態を変化させて入出力させることによって、ターゲットマシン上でのペリフェラルへの操作を実行マシン上でのペリフェラルへの操作に変換する請求項 1 に記載のシステムエミュレータ。

【請求項 5】 前記ペリフェラルエミュレータは、内部状態をユーザーに通知する手段を有する請求項 2～4 のいずれかに記載のシステムエミュレータ。

【請求項 6】 前記ペリフェラルエミュレータは、外部からの入力もしくは時間経過によって内部状態を変化させる手段を有する請求項 2～4 のいずれかに記載のシ

テムエミュレータ。

【請求項 7】 前記ペリフェラルエミュレータは、ユーザーからの操作によって内部状態の変更もしくはレジスタへの書きこみを行う手段を有する請求項 2～4 のいずれかに記載のシステムエミュレータ。

【請求項 8】 前記ペリフェラルエミュレータは、内部状態の変化を他のペリフェラルエミュレータに通知して、その内部状態を変化させる手段を有する請求項 2～4 のいずれかに記載のシステムエミュレータ。

【請求項 9】 前記ペリフェラルエミュレータは、内部状態の変化をインストラクションセットシミュレータに通知して、インストラクションセットシミュレータのプログラムの実行状態を変化させる手段を有する請求項 2～4 のいずれかに記載のシステムエミュレータ。

【請求項 10】 前記ペリフェラルエミュレータは、実行マシン上で外部の機器からの入力を受け付けるペリフェラルを制御するペリフェラルエミュレータであり、入力に起因した内部状態の変化の通知を他のペリフェラル及びインストラクションセットシミュレータに通知できない場合に入力したデータを保留できる手段を有する請求項 8 又は 9 に記載のシステムエミュレータ。

【請求項 11】 前記ペリフェラルエミュレータは、実行マシン上で外部の機器へデータの出力を行うペリフェラルを制御するペリフェラルエミュレータであり、外部からデータの出力を要求されても他のペリフェラルエミュレータもしくはインストラクションセットシミュレータが停止もしくは応答するのに十分な速度を有していないためにデータが送信できない場合に、内部状態、他のペリフェラルエミュレータ、インストラクションセットシミュレータの内部状態から独自に出力すべきデータを生成して出力する手段を有する請求項 8 又は 9 に記載のシステムエミュレータ。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、ターゲットマシン用のプログラムを実行マシン上で動作させるシステムエミュレータに関し、特に、ターゲットマシン上での実行環境とエミュレータ上での実行環境との差異を吸収して、プログラムの開発効率を向上させることができるシステムエミュレータに関する。

【0002】

【従来の技術】 ターゲットマシン上で動作するプログラムの開発段階においては、インサーキットエミュレータ上にプログラムをロードして実行し、実行状態の制御および監視を行うことにより、デバッグが行われている。インサーキットエミュレータには、ターゲットマシンと同様に、CPU、メモリ、I/O コントローラ、2 次記憶装置等のハードウェア資源が、ペリフェラル（周辺機器）として設けられており、各ペリフェラルのアーキテクチャも、ターゲットマシンと同様になっている。

【0003】このように、インサーキットエミュレータは、ターゲットマシンと同様のハードウェア資源、アーキテクチャを有するため、ターゲットマシン用のプログラムをそのまま実行することができる。

【0004】しかしながら、このようなインサーキットエミュレータによる方式は、ターゲットマシンの動作を完全にエミュレートすることができるが、ロード時間が長く、プログラムにバグがあった場合に動作が不安定となるおそれがある等の問題がある。また、エミュレートの実行には、専用のハードウェアが必要であるために、プログラムの開発者全員に対して、専用のハードウェアを準備することは、コスト的に不利であるという問題もある。

【0005】このため、パーソナルコンピュータPC、EWS等の汎用的なマシン上に、ターゲットマシンの動作を模倣するエミュレータを設けて、このエミュレータにて、ターゲットマシン用のプログラムを実行させることにより、ロード時間を短縮するとともに、コストを低減する手法が提案されている。

【0006】例えば、特開平6-149621号公報には、高級言語で書かれたターゲットマシン用のプログラムと、ターゲットマシン上にペリフェラルを制御するライブラリを実行マシン上に模倣するライブラリとに基づいて、実行マシン上に、直接実行可能な実行イメージを作成するエミュレータが開示されている。

【0007】また、パーソナルコンピュータやEWS上に、ターゲットマシンの命令セットを理解して実行するインストラクションセットシミュレータも提案されている。例えば、特表平11-505645号公報には、インストラクションセットシミュレータとハードウェアのシミュレータとを結合することにより、ペリフェラルを含めた実行環境を実行マシン上に実現するシステムシミュレータが開示されている。

【0008】

【発明が解決しようとする課題】しかしながら、特開平6-149621号公報に開示されたエミュレータでは、実行時の命令セットとは異なる命令セットにてエミュレートするとともに、ターゲットマシンにおける割り込み等の低レベルの機能を模倣することができないために、エミュレータ上とターゲットマシン上での実行環境の差異が大きいという問題がある。

【0009】これに対し、インストラクションセットシミュレータでは、ターゲットマシンにおける割り込み等の低レベル機能の模倣が可能であり、前述のような問題は起らない。しかし、仮想的な実行環境を実行マシン上で整える方式のため、ペリフェラルの模倣もまた仮想的になる。その結果、I/Oコントローラに現実の機器を接続することができず、ターゲットマシンの実行環境との間に差異が生じることになる。

【0010】また、特表平11-505645号公報に

開示されたシステムシミュレータでは、すべての動作をシミュレータ上で実行するため、例えば、シリアルポートにターミナルを接続するといったエミュレーションができないという制限があり、この点において、ターゲットマシンの実行環境との間に差異が生じることになる。

【0011】このように、エミュレータとターゲットマシンとの実行環境の差異が存在することによって、プログラムの開発効率が低下するおそれがある。

【0012】本発明は上記従来の問題を解決するものであり、その目的は、インストラクションセットシミュレータから読み出されるペリフェラルエミュレータを、実行マシンに存在するペリフェラルに結びつけることにより、ターゲットマシンと同様の実行環境を実行マシン上にて実現することができるシステムエミュレータを提供することにある。

【0013】

【課題を解決するための手段】本発明のシステムエミュレータは、実行マシン上に、命令セットの異なるターゲットマシン用のプログラムを動作させるシステムエミュレータであって、ターゲットマシンの命令セットを実行マシン上に仮想的に実行するインストラクションセットシミュレータと、ターゲットマシン上に異なる機能をそれぞれ実行する複数のペリフェラルの動作を、実行マシン上にそれぞれ模倣するペリフェラルエミュレータとを具備し、各ペリフェラルエミュレータは、模倣するペリフェラルの機能、構造の複雑度に応じた方式によって、ターゲットマシン上での各ペリフェラルに対する操作を、実行マシン上での各ペリフェラルの操作にそれぞれ変換するようになっていることを特徴とする。

【0014】前記各ペリフェラルエミュレータは、インストラクションセットシミュレータからペリフェラルのレジスタへのアクセスがあった場合に、そのアクセスの内容を解析し、対応する実行マシン上のペリフェラルのレジスタにアクセスを行うことによって、ターゲットマシン上でのペリフェラルへの操作を実行マシン上でのペリフェラルへの操作に変換する。

【0015】前記ペリフェラルエミュレータは、インストラクションセットシミュレータからペリフェラルのレジスタへのアクセスがあった場合に、そのアクセスの内容を解析し、ペリフェラルエミュレータの各種設定などの内部状態を変化させるとともに、実行マシン上のペリフェラルにアクセスして、実行マシン上でのペリフェラルの状態を変化させて入出力させることによって、ターゲットマシン上でのペリフェラルへの操作を実行マシン上でのペリフェラルへの操作に変換する。

【0016】前記ペリフェラルエミュレータは、ペリフェラルを扱う同一インターフェースのライブラリを、ターゲットマシン用のライブラリとは別に有しており、プログラムを実行する際に、プログラムからライブラリを呼び出し、ライブラリによってペリフェラルエミュレー

(4)

5

タの内部状態を変化させ、実行マシン上のペリフェラルにアクセスし、実行マシン上でのペリフェラルの状態を変化させて入出力させることによって、ターゲットマシン上でのペリフェラルへの操作を実行マシン上でのペリフェラルへの操作に変換する。

【0017】前記ペリフェラルエミュレータは、内部状態をユーザーに通知する手段を有する。

【0018】前記ペリフェラルエミュレータは、外部からの入力もしくは時間経過によって内部状態を変化させる手段を有する。

【0019】前記ペリフェラルエミュレータは、ユーザーからの操作によって内部状態の変更もしくはレジスタへの書きこみを行う手段を有する。

【0020】前記ペリフェラルエミュレータは、内部状態の変化を他のペリフェラルエミュレータに通知して、その内部状態を変化させる手段を有する。

【0021】前記ペリフェラルエミュレータは、内部状態の変化をインストラクションセットシミュレータに通知して、インストラクションセットシミュレータのプログラムの実行状態を変化させる手段を有する。

【0022】前記ペリフェラルエミュレータは、実行マシン上で外部の機器からの入力を受け付けるペリフェラルを制御するペリフェラルエミュレータであり、入力に起因した内部状態の変化の通知を他のペリフェラル及びインストラクションセットシミュレータに通知できない場合に入力したデータを保留できる手段を有する。

【0023】前記ペリフェラルエミュレータは、実行マシン上で外部の機器へデータの出力を行うペリフェラルを制御するペリフェラルエミュレータであり、外部からデータの出力を要求されても他のペリフェラルエミュレータもしくはインストラクションセットシミュレータが停止もしくは応答するのに十分な速度を有していないためにデータが送信できない場合に、内部状態、他のペリフェラルエミュレータ、インストラクションセットシミュレータの内部状態から独自に出力すべきデータを生成して出力する手段を有する。

【0024】本発明のシステムエミュレータでは、インストラクションセットシミュレータは、ターゲットマシン上で動作するプログラムを実行し、そのプログラムの内容に従ってペリフェラルを操作する。ペリフェラルへの操作は、ペリフェラルの持つレジスタへのアクセスとして処理される。このアクセスは、システムエミュレータ上では、ペリフェラルエミュレータへと回され、アクセスを受けたペリフェラルエミュレータは、アクセス内容を解析し、要求された処理を行う。

【0025】一般に、模倣を行うべきペリフェラルの構造、動作の複雑度は、それぞれのペリフェラルによって異なっており、また、ターゲットマシンに存在するペリフェラルがすべて実行マシン上で用意できるとは限らない。そのため、本発明のシステムエミュレータでは、機

能毎にペリフェラルエミュレータを分割して、ターゲットマシン上のペリフェラルの構造、動作に合わせた方式により、ペリフェラルの模倣を行う。ペリフェラルエミュレータにおける模倣の方式としては、例えば以下の3つを使い分ける。

【0026】ターゲットマシン上、実行マシン上に存在するペリフェラルの操作方法、すなわちレジスタへのアクセス方法が同じであり、しかも、実行マシン上のペリフェラルのレジスタをペリフェラルエミュレータから直接操作できる場合は、インストラクションセットシミュレータからペリフェラルのレジスタへアクセスがあった場合に、アクセスされたアドレス、I/Oポートを実行マシン上のアドレス、I/Oポートに変換し、実行マシン上のペリフェラルのレジスタにアクセスする。

【0027】この方式を採用できない場合は、インストラクションセットシミュレータからペリフェラルのレジスタへアクセスがあった場合に、アクセスの内容からターゲットマシン上のペリフェラルの状態遷移を予測し、実行マシン上で対応するペリフェラルの状態を変化させ、実行マシン上のペリフェラルの状態とターゲットマシン上のペリフェラルの状態とを一致させる。

【0028】レジスタアクセスの方法、動作が複雑な場合は、ペリフェラルを扱う同一インターフェースのライブラリをターゲットマシン用およびエミュレータ用にそれぞれ用意する。エミュレータ上で実行する場合は、プログラムとエミュレータ用のライブラリをリンクし、エミュレータ用のライブラリがペリフェラルエミュレータの内部状態を変化させ、必要に応じて実行マシン上のペリフェラルにアクセスし、実行マシン上のペリフェラルの状態を変化させて入出力を行わせることにより、ペリフェラルの模倣を行う。ターゲットマシン上で実行する場合は、このライブラリをターゲットマシン用のペリフェラルを操作するライブラリに差し替えることによって対応する。

【0029】

【発明の実施の形態】以下、図面を参照しながら本発明の実施の形態を説明する。

【0030】本実施形態のシステムエミュレータは、ターゲットマシンのエミュレータを、ウィンドウシステムが機能しているパーソナルコンピュータPCを実行マシンとして、このパーソナルコンピュータPC上にて実現するようになっている。まず、ターゲットマシンの構成を図1に示す。

【0031】図1に示すように、ターゲットマシン10は、CPU11と、割り込みコントローラINTC12と、シリアルポートコントローラUART13と、液晶表示装置LCD14と、座標入力装置PD15と、メインメモリ16とを、ペリフェラルとして備えている。

【0032】CPU11は、メインメモリ16および各ペリフェラル12～15と、共通のバスを通じて接続さ

れている。メインメモリ16と各ペリフェラル12～15のレジスタは、図2に示すように、同一アドレス空間上に割り当てられている。

【0033】割り込みコントローラINTC12は、2チャンネルからの割り込み入力が可能になっている。この割り込みコントローラINTC12は、メモリマップ上から複数のレジスタを参照することができるようになっており、各レジスタにアクセスすることにより、2つのチャンネルからの割り込み入力に関して、割り込みの許可および禁止の設定、割り込みの発生状況の調査を行く。割り込みコントローラINTC12は、割り込み入力ピンに加わる電圧が下がったローレベル時に割り込みが発生したと判断し、その割り込みの許可および禁止を調べ、割り込みが許可されている場合に、CPU11の割り込み入力ピンに接続されている出力ピンに印加される電圧をローレベルとして、CPU11に対して割り込みを通知する。

【0034】シリアルポートコントローラUART13は、一般的なパーソナルコンピュータが有するシリアルポートコントローラと同等の機能を有しており、通信速度、ストップビット、パリティ、フロー制御を変更できるようになっている。このシリアルポートコントローラUART13も、割り込みコントローラINTC12と同様に、内部状態の変更、参照等が可能なレジスタをメモリマップ上に有しており、前述した各種設定、送信データの書き込み、受信データの読み込み、割り込みの許可および禁止の設定等が可能になっている。シリアルポートコントローラUART13の割り込み信号は、データ受信、データ送信の完了、エラー発生、フロー制御信号の変化等により発生するようになっており、発生した割り込み信号は、割り込みコントローラINTC12の割り込み第1チャンネルに割り当てられている。

【0035】液晶表示装置LCD14は、表示用のメモリが、メモリマップから直接、参照および変更することができるようになっており、この内容を変更することによって表示内容を変更することができる。メモリの1ビットの「0」および「1」が、液晶表示装置LCD14の画面上における1ピクセルの「白」および「黒」に、それぞれ対応している。

【0036】座標入力装置PD15は、感圧式のフィルムを有しており、このフィルムを押圧することにより割り込み信号が発生し、メモリマップ上に配置されたレジスタに基づいて、押圧された点の2次元座標を出力することができる構造になっている。座標入力装置PD15の割り込み信号は、割り込みコントローラの第2チャンネルに割り当てられている。この座標入力装置PD15の感圧式フィルムは、透明であって、液晶表示装置LCD14上に重ねられており、液晶表示装置LCD14上の座標位置が、感圧式フィルムにて押圧される2次元の座標位置に対応している。

【0037】本実施形態のシステムレミュレータは、このような構成のターゲットマシン10のエミュレータを、ウィンドウシステムが機能しているパーソナルコンピュータPC上にて実現する。このシステムエミュレータの全体の構成を図3に示す。

【0038】図3に示すように、システムエミュレータ20は、インストラクションセットシミュレータISS21と、ターゲットマシン10の各ペリフェラル12～15の模倣をそれぞれ行う複数のペリフェラルエミュレータが設けられている。各ペリフェラルエミュレータとしては、割り込みコントローラエミュレータINTCE22と、シリアルポートコントローラエミュレータUART23と、座標入力装置エミュレータPDE25と、液晶表示装置エミュレータLCDE24と、パーソナルコンピュータPCのメモリへの読み込みおよび書き込みをシステムエミュレータ20上で実現するメモリアクセスエミュレータMAE26と、インストラクションセットシミュレータISS21からメモリマップ上へのアクセスを、各ペリフェラルエミュレータに振り分けるI/OマネージャIOM27とを備えている。

【0039】システムエミュレータ20は、パーソナルコンピュータPCによって動作するデバugg30により制御される。デバugg30は、実行、一時停止、リセットなどの制御を、インストラクションセットシミュレータISS21および各ペリフェラルエミュレータに対して行う。

【0040】このような構成のシステムエミュレータの動作について、3つの具体例をあげて説明する。

【0041】まず、全てのペリフェラルエミュレータPEを実現する第1の具体例について説明する。

【0042】インストラクションセットシミュレータISS21は、ターゲットマシン10のCPU11の動作を、実行マシンであるパーソナルコンピュータPC上にて模倣するためのソフトウェアである。インストラクションセットシミュレータISS21の構成を図4に示す。

【0043】インストラクションセットシミュレータISS21は、現実のターゲットマシン10のCPU11と同様に、内部に複数の汎用レジスタGR0～GR3と、プログラムカウンタPrCと、フラグレジスタFRとを有している。汎用レジスタGR0～GR3には、演算数および演算結果が記録される。プログラムカウンタPrCには、次に実行すべき命令が記録されたメモリのアドレスが記録されている。フラグレジスタFRには、演算結果の正および負、演算結果が0か否か、演算が桁あふれ（オーバーフロー）を起こしたかなどの情報が演算後に記録される。

【0044】現実のターゲットマシン10のCPU11は、メモリバス、アドレスバスを介してメインメモリ16におけるメモリマップ上の特定のアドレスからデータ

の読み込みおよび書き込みを行う。インストラクションセットシミュレータISS21は、そのCPU11に代わって、I/OマネージャIOM27との間でメモリマップへアクセスするための関数インターフェースを有している。

【0045】I/OマネージャIOM27は、インストラクションセットシミュレータISS21からのアクセスを、各ペリフェラルエミュレータPEに振り分ける機能を有している。I/OマネージャIOM27の動作を、図5に示すフローチャートに基づいて説明する。

【0046】インストラクションセットシミュレータISS21は、メモリマップにアクセスしたい場合、I/OマネージャIOM27に対して、データの存するアドレスと、読み込み、書き込みのいずれのためにアクセスを行うかを示すアクセス内容とを出力する。

【0047】I/OマネージャIOM27は、インストラクションセットシミュレータISS21からアドレスおよびアクセス内容を得ると、アクセスされるアドレスが、メモリマップ内の正常なアドレスを示しているかを検証する(ステップS501)。正常なアドレスを示していない場合には、インストラクションセットシミュレータISS21にアクセスが失敗したことを通知して処理を終了する。正常なアドレスを示している場合には、得られたアドレスから、メモリアクセスエミュレータMAE26、もしくは、他のペリフェラルエミュレータPEへのアクセスかを判断する(ステップS502)。

【0048】アクセスすべきペリフェラルエミュレータPEが判明すれば、インストラクションセットシミュレータISS21からの要求が読み込みであるか書き込みであるかを判断する(ステップS503)。読み込みである場合は、対応するメモリアクセスエミュレータMAE26もしくは他のペリフェラルエミュレータPEに対してデータの読み込み要求を行う(ステップS504)。書き込みである場合は、対応するメモリアクセスエミュレータMAE26もしくはペリフェラルエミュレータPEに対してデータの書き込み要求を行う(ステップS505)。

【0049】アクセス要求を受けたメモリアクセスエミュレータMAE26、あるいは、他のペリフェラルエミュレータPEは、アクセスが読み込み要求であれば、指定されたアドレスに対応するペリフェラルエミュレータPEのメモリあるいはレジスタのデータ内容を読み出す。そして、I/OマネージャIOM27は、読み出されたデータを、インストラクションセットシミュレータISS21に出力する。これに対して、メモリアクセスエミュレータMAE26あるいは他のペリフェラルエミュレータPEに対するアクセスが書き込み要求の場合には、指定されたアドレスに対応するメモリアクセスエミュレータMAE26もしくはペリフェラルエミュレータPEのレジスタに対してデータを書き込む。

【0050】インストラクションセットシミュレータISS21の動作は、現実のターゲットマシン10のCPU11と全く同様になっている。すなわち、インストラクションセットシミュレータISS21は、デバッグ30からリセット要求を受けると、プログラムカウンタPrCを「0」にクリアする。その後、インストラクションセットシミュレータISS21は、以下の動作を繰り返すことによりプログラムを実行する。

【0051】インストラクションセットシミュレータISS21は、プログラムカウンタPrCにて示されるアドレスに収められたメモリのデータを読み出す。インストラクションセットシミュレータISS21は、読み出されたデータを、演算の種類を指定する命令として扱い、命令の種類を解析して実行する。

【0052】例えば、読み出されたデータの命令が、レジスタ間での演算を実行する場合には、指定されたレジスタ間での演算を実施し、演算結果を、指定されたレジスタGR0~GR3のいずれかに保存し、フラグレジスタFRの内容を更新する。読み出されたデータの命令が、レジスタGR0~GR3からメモリアクセスエミュレータMAE26もしくはペリフェラルエミュレータPEのレジスタへのデータ転送を要求する場合には、I/OマネージャIOM27に対して、データの書き込み要求を行い、指定されたレジスタの内容を、指定されたアドレスに対応するメモリアクセスエミュレータMAE26もしくはペリフェラルエミュレータPEのレジスタに書き込む。

【0053】さらに、読み出されたデータの命令が、メモリアクセスエミュレータMAE26もしくはペリフェラルエミュレータPEのレジスタから、インストラクションセットシミュレータISS21のレジスタへのデータ転送を要求する場合には、I/OマネージャIOM27に対して、データの読み込み要求を行い、読み出されたデータの命令によって指定されたメモリアクセスエミュレータMAE26もしくはペリフェラルエミュレータPEのアドレスの内容を、指定されたインストラクションセットシミュレータISS21のレジスタに読み込む。命令の実行が終了すれば、プログラムカウンタPrCの値をインクリメントし、次の命令を読み込む。

【0054】このような動作を繰り返すことによって、インストラクションセットシミュレータISS21はプログラムを実行する。

【0055】メモリアクセスエミュレータMAE26は、システムエミュレータ20上のメモリと、実行マシンであるパーソナルコンピュータPC上のメモリとを、1対1に対応させ、インストラクションセットシミュレータISS21からメモリへのアクセスがあると、これを、パーソナルコンピュータPC上のメモリへのアクセスに変換する。このようにして、ターゲットマシン10上でのメインメモリ16の動作を、実行マシンであるパ

ーソナルコンピュータPC上にて模倣する。

【0056】割り込みコントローラエミュレータINTCE22は、インストラクションセットシミュレータISS21から参照可能なレジスタを、システムエミュレータ20のメモリマップ上に有しており、割り込みコントローラエミュレータINTCE22の内部状態、すなわち割り込みの許可、禁止、割り込みの発生状態の変更、参照等を行うことができる。

【0057】図6は、インストラクションセットシミュレータISS21からアクセスされた場合の割り込みコントローラエミュレータINTCE21の動作説明のためのフローチャートである。割り込みコントローラエミュレータINTCE22は、指定されたアドレスが正常であるかを判定し(ステップS601)、正常でなければI/OマネージャIOM27に、アクセスが失敗したことを通知し、割り込みコントローラエミュレータINTCE21の処理を終了する。指定されたアドレスが正常であれば、指定されたアクセスが、読み込み要求か書き込み要求かを判定する(ステップS602)。指定されたアクセスが、読み込み要求の場合には、内部状態を設定データとしてインストラクションセットシミュレータISS21に返す(ステップ603)。これに対して、指定されたアクセスが、書き込み要求の場合には、設定データを内部に保存する(ステップS604)。割り込みコントローラエミュレータINTCE22は、割り込みの通知を受けるための関数エントリを有しており、他のペリフェラルエミュレータPEは、これらを出すことによって割り込み発生を通知することができる。

【0058】割り込みコントローラエミュレータINTCE22に対して割り込み発生が通知されると、割り込みコントローラエミュレータINTCE22は割り込みの発生要因を判別し(ステップS605)、内部にこの情報を保存する(ステップS606)。次に、内部状態を確認し(ステップS607)、割り込みが許可されている場合には、インストラクションセットシミュレータISS21に割り込みの発生を通知する(ステップS608)。

【0059】シリアルポートコントローラエミュレータUARTE23も、インストラクションセットシミュレータISS21から参照可能なレジスタを、システムエミュレータ20のメモリマップ上に有しており、このレジスタへアクセスすることにより、シリアルポートコントローラエミュレータUARTE23の内部状態の変更、すなわち通信速度、ストップビット、パリティ、フロー制御などの各種設定、送信データの書き込み、受信データの読み込み、割り込みの許可および禁止の設定等が行える。

【0060】図7は、シリアルポートコントローラエミュレータUARTE23のレジスタがアクセスを受けた

場合のシリアルポートコントローラエミュレータUARTE23の動作説明のためのフローチャートである。

【0061】シリアルポートコントローラエミュレータUARTE23のレジスタにアクセスがあった場合、シリアルポートコントローラエミュレータUARTE23は、アクセスされたアドレスの正当性を検証する(ステップS701)。アクセスされたアドレスが正常の場合には、読み込み、書き込みいずれのいずれのアクセスかを判断する(ステップS702)。

【0062】アクセスが読み込みの場合、シリアルポートコントローラエミュレータUARTE23は、アクセスされたアドレスから、受信データを受け取るレジスタへのアクセスか、設定データを取得するレジスタへのアクセスかを判断する(ステップS703)。受信データを受け取るレジスタからの読み込みの場合は、シリアルポートコントローラエミュレータUARTE23内の受信データバッファからデータを読み込み、I/OマネージャIOM13へ出力する(ステップS704)。

【0063】受信データバッファは、実行マシンであるパーソナルコンピュータPC上のシリアルポートから読み込んだデータを保存するためのバッファである。シリアルポートコントローラエミュレータUARTE23は、定期的にパーソナルコンピュータPCのシリアルポートがデータを受信していないかを監視しており、データを受信している場合には、受信したデータを受信データバッファに保存する。このような受信データバッファが設けられていることにより、システムエミュレータ20の動作が遅く、受信データの取得が間に合わずパーソナルコンピュータPCのシリアルポートがオーバーフローするというエラーの発生を防止することができる。

【0064】アクセスが読み込みであり、しかも、設定データを取得するレジスタへのアクセスの場合には、そのアクセスがフロー制御の状態を取得するものであるかを判定する(ステップS705)。アクセスがフロー制御の状態を取得するものである場合には、パーソナルコンピュータPCのシリアルポートのフロー制御の状態を取得してI/Oマネージャ27に出力する(ステップS707)。フロー制御の状態を取得するものでない場合には、内部に保存している設定データをI/Oマネージャ27に出力する(ステップS706)。

【0065】アクセスが書き込みの場合、アクセスされたアドレスから、送信データを書きこむレジスタへのアクセスか、設定を行うレジスタへのアクセスかを判断する(ステップS708)。そして、送信データを書きこむレジスタへのアクセスの場合には、書き込まれたデータをパーソナルコンピュータPC上のシリアルポートから出力する(ステップS709)。設定を行うレジスタへのアクセスの場合には、そのアクセスがフロー制御の状態を設定するものであるかを判定する(ステップS710)。アクセスがフロー制御の状態を設定するもので

ある場合には、パーソナルコンピュータPCのフロー制御の状態を設定する(ステップS713)。アクセスがフロー制御の状態を設定するものでない場合には、シリアルポートコントローラエミュレータUARTE23の内部に設定データを保存し(ステップS711)、変更された設定に基づいて、パーソナルコンピュータPC上のシリアルポートコントローラの設定を変更する(ステップS712)。

【0066】シリアルポートコントローラエミュレータUARTE23は、定期的にパーソナルコンピュータPC上のシリアルポートのフロー制御の入力を監視しており、状態の変化があった場合には、内部状態も、これに合わせて変更する。また、インストラクションセットシミュレータISS21がデバッグ30により停止させられている場合には、フロー制御の状態をデータ受信不可にして、通信相手に、データ通信が行えないことを通知する。シリアルポートコントローラエミュレータUARTE23等の各ペリフェラルエミュレータPEは、図示しない経路によって、直接、インストラクションセットシミュレータISS21の状態を、外部からデータ入力があった場合等に確認する。インストラクションセットシミュレータISS21が停止状態の場合、あるいは、十分な応答が得られない場合は、シリアルポートコントローラエミュレータUARTE23は、パーソナルコンピュータPC上のシリアルポートのフロー制御の状態を変化させ、パーソナルコンピュータPC上のシリアルポートに接続された機器に対して、現在、データを受け付けることができないことを通知する。

【0067】シリアルポートコントローラエミュレータUARTE23は、パーソナルコンピュータPC上のシリアルポートの設定変更および入出力を、パーソナルコンピュータPCにおけるオリジナルソフトOSのアプリケーション・プログラム・インターフェイスAPIを介して行う。

【0068】シリアルポートコントローラエミュレータUARTE23は、割り込みを発生させることが可能である。割り込み発生要因はいくつかあり、内部に保存している設定情報により、割り込み発生が許可されている場合には、それぞれについて判定を行った後、割り込みコントローラエミュレータINTCE21に割り込みを通知する。

【0069】送信完了時の割り込みが許可されている場合は、パーソナルコンピュータPC上のシリアルポートヘデータを送信し、その送信が完了したとき、割り込みコントローラエミュレータINTCE21に割り込みを通知する。データ受信による割り込みが許可されている場合は、パーソナルコンピュータPC上のシリアルポートを定期的に監視したときにデータが到着していれば、割り込みを発生させる。エラー発生割り込みが許可されている場合には、パーソナルコンピュータPC上のシ

リアルポートへのアクセス時にエラーが発生していないことを確認し、発生している場合は、その原因を取得し、内部にエラー情報を記録した後、割り込みを発生させる。フロー制御信号の変化による割り込みが許可されている場合は、パーソナルコンピュータPC上のシリアルポートのフロー制御の入力状態が変化した時に、内部状態を変化させると共に、割り込みを発生させる。

【0070】液晶表示装置エミュレータLCDE24は、表示用のビットマップを保持するメモリをシステムエミュレータ20のメモリマップ上に有しており、このメモリの内容を、ウィンドウシステムのアプリケーション・プログラム・インターフェイスAPIを使用してパーソナルコンピュータPCの画面上に描画する。

【0071】座標入力装置エミュレータPDE24は、ウィンドウシステムからのマウスのイベントを待っており、座標入力装置エミュレータPDE24が表示する画面イメージ上にてマウスがドラッグされた場合、インストラクションセットシミュレータISS21に割り込みを発生させる。また、座標入力装置エミュレータPDE24は、メモリマップ上にレジスタを配置しており、レジスタからの読み込みがあった場合は、マウスがドラッグされた座標をインストラクションセットシミュレータISS21へ出力する。

【0072】これらのペリフェラルエミュレータPEは、すべてデバッグ30からの要求により、内部状態の参照および変更が可能になっており、デバッグ30から内部状態の読み込みの要求があった場合は、各ペリフェラルエミュレータPEは、保持している内部状態をデバッグ30に出力する。デバッグ30は、入力された内部状態を、デバッグ30の操作者に対して、グラフィカル・ユーザー・インターフェイスGUIを用いて表示する機能を有する。このグラフィカル・ユーザー・インターフェイスGUIが操作者によって操作されると、デバッグ30からペリフェラルエミュレータPEに内部状態の変更要求が出力され、これを受けたペリフェラルエミュレータPEは、操作者によって変更された内部状態を保存する。この機能により、内部状態の表示および変更が可能になる。ペリフェラルエミュレータPEがシリアルポートコントローラエミュレータUARTE23の場合には、通信速度、ストップビット、パリティ、フロー制御などの設定、データ送受信状態の表示および変更が可能である。

【0073】このように、システムエミュレータ20には、実行マシンであるパーソナルコンピュータPC上の各ペリフェラルを模倣するペリフェラルエミュレータPEがそれぞれ作成されて、搭載されている。しかも、液晶表示装置エミュレータLCDE24と座標入力装置エミュレータPDE24とによって、ユーザインターフェースが実現されており、シリアルポートコントローラエミュレータUARTE23によって外部機器との通信

が、ターゲットマシン10と同様にシステムエミュレータ20上にて実現されている。

【0074】次に、上記構成のシステムエミュレータ20によってアプリケーションプログラムを実行させる場合のシステムエミュレータ20の動作について説明する。アプリケーションプログラムとして、座標入力装置PD15から座標入力によって、入力された座標に対応する液晶表示装置LCD14のピクセルを黒にするという簡単な「描画アプリケーション」と、パーソナルコンピュータPCのシリアルポートから座標データを受け取って、指定された座標に対応する液晶表示装置LCD24のピクセルを黒にするという簡単な「通信アプリケーション」とについて説明する。

【0075】まず「描画アプリケーション」について説明する。描画アプリケーションのフローチャートを、図8に示す。描画アプリケーションは、実行直後に割り込みコントローラINTC12の割り込み第2チャンネル、すなわち座標入力装置PD15からの割り込みを許可状態にする(ステップS801)。その後、描画アプリケーションは待機状態に移行する。割り込みコントローラINTC12の割り込み第2チャンネルに割り込みが発生すると、割り込みコントローラINTC12を調査する(ステップS802)。そして、割り込み要因を判別し(ステップS803)、割り込み要因が座標入力装置PD15からの座標入力の場合には、割り込み要因をクリアする(ステップS804)。

【0076】次に、描画アプリケーションは、座標入力装置PD15の座標を取得するレジスタからデータを読み出す(ステップS805)。このデータは、座標入力装置PD15の感圧フィルムが押圧された座標を表している。描画アプリケーションは、座標入力装置PD15の感圧フィルムが押圧された座標に対応する液晶表示装置LCD14の表示用メモリのビットに「1」を設定する(ステップS806)。

【0077】このようにして、描画アプリケーションは、座標入力装置PD15の感圧フィルムが押圧された座標に対応する液晶表示装置LCD15のピクセルを、「黒」にする。

【0078】描画アプリケーション実行時のエミュレータの動作は以下になる。インストラクションセットシミュレータISS21に描画アプリケーションの実行を要求すると、インストラクションセットシミュレータISS21は、I/OマネージャIOM27を介して、メモリアクセスエミュレータMAE26から、順次、アプリケーションプログラムを読み出し、実行する。

【0079】描画アプリケーションは、最初に割り込みコントローラINTC12の割り込み第2チャンネルからの割り込みを許可状態にする。この動作は、CPU11が割り込みコントローラINTC12の割り込み設定

を行うレジスタにデータを書き込むことにより行われる。インストラクションエミュレータ20では、次のような動作により設定が行われる。

【0080】インストラクションセットシミュレータISS21は、割り込みコントローラエミュレータINTCE22の割り込み設定を行うレジスタが存在するメモリマップ上のアドレスに対して、データを書き込む要求を、I/OマネージャIOM27に出力する。I/OマネージャIOM27は、要求されているアドレスから、この書き込み要求が割り込みコントローラエミュレータINTCE22に対するものであることを判断し、割り込みコントローラエミュレータINTCE22にデータの書き込み要求を出力する。割り込みコントローラエミュレータINTCE22は、要求されたアドレスおよびデータが、割り込み第2チャンネルを割り込み許可状態にするものであることを解析し、内部に保持している割り込み第2チャンネルの割り込み許可状態を、許可に設定する。

【0081】このようにして、描画アプリケーションからペリフェラルエミュレータPEのレジスタへのアクセスが処理される。座標入力装置エミュレータPDE25、液晶表示装置エミュレータLCDE24、シリアルポートコントローラエミュレータUARTE23等、他のペリフェラルエミュレータPEへのアクセスも同様の手法で処理される。

【0082】割り込みコントローラの設定が終わると、描画アプリケーションは割り込みを待つ待機状態になる。

【0083】この間、座標入力装置エミュレータPDE25は、ウィンドウシステムからのマウスイベントを待っている。座標入力装置エミュレータPDE25は、ウィンドウシステムからマウスのボタンが押圧されたという情報を受け取ると、内部状態が変化したものとして、マウスのボタンの押圧状態と、押圧位置に対応した座標とを、座標入力装置エミュレータPDE25の内部に記録して、割り込みコントローラエミュレータINTCE22に割り込みの発生を通知する。

【0084】通知を受けた割り込みコントローラエミュレータINTCE22は、座標入力装置エミュレータPDE25からの割り込みの許可状態を調査する。この場合、アプリケーションにより割り込みコントローラの初期設定が終了していると、座標入力装置エミュレータPDE25からの割り込み状態が許可に設定されているために、インストラクションセットシミュレータISS21に対して割り込みの発生を通知する。この時、割り込みコントローラエミュレータINTCE22の内部状態は、割り込みの発生状態に変更される。

【0085】割り込みの通知を受けたインストラクションセットシミュレータISS21は、プログラムの実行を割り込み発生時の処理に切り替える。割り込み発生時

には、描画アプリケーションは、割り込みコントローラを調査して割り込み要因を調査する。この動作は、システムエミュレータ20では、割り込み許可状態の設定と同様に行われる。割り込み要因を調査するために、インストラクションセットシミュレータISS21は、I/OマネージャIOM27を介して、割り込みコントローラエミュレータINTCE22に割り込み要因を示すレジスタからのデータの読み込み要求を行う。

【0086】これを受けた割り込みコントローラエミュレータINTCE22は、内部に保存していた割り込み要因を出力する。出力されたデータから、割り込み要因が座標入力装置PD15にあるかが判定される。割り込み要因が座標入力装置PD15にある場合は、描画アプリケーションは割り込みをクリアした後、座標入力装置PD15にて入力操作された座標を保持するレジスタを読み出す。この要求を受けた座標入力装置エミュレータPDE25は、内部に保持している座標を、インストラクションセットシミュレータISS21へ出力する。

【0087】インストラクションセットシミュレータISS21上で動作する描画アプリケーションは、こうして座標入力装置PD15の入力操作状態、入力された座標を得る。

【0088】座標入力装置PD15に対して入力された座標が得られると、次に、描画アプリケーションは液晶表示装置LCD14に対して点を描画する。アプリケーションは、座標入力装置PD15に対して入力操作された座標から、変更すべき液晶表示装置LCD14の表示用メモリのビットを計算し、このビットを「1」に設定することによって、点を描画する。この動作は、システムエミュレータ20では、インストラクションセットシミュレータISS21が、I/OマネージャIOM27を介して液晶表示装置エミュレータLCDE24における表示メモリの特定のビットを「1」に設定し、表示メモリのデータが変更されたことを検知した液晶表示装置エミュレータLCDE14がウィンドウシステムのアプリケーション・プログラム・インターフェースAPIを呼び、表示メモリのデータをパーソナルコンピュータPCの画面に表示させることによって行われる。

【0089】以上のようにして、描画アプリケーションが、システムエミュレータ20上で実行される。

【0090】次に「通信アプリケーション」について、図9に示す通信アプリケーションのフローチャートに基づいて説明する。

【0091】通信アプリケーションは、実行直後に通信のパラメータ、割り込みの許可状態を設定する。最初に、通信パラメータが設定される。通信パラメータの設定は、シリアルポートコントローラUART13における通信パラメータを設定するレジスタに設定データを書きこむことにより行われる。通信パラメータとして、通信速度、パリティの有無等の設定、データ受信完了時お

よびデータ送信完了時に割り込みを発生させる設定を行う（ステップS901）。

【0092】シリアルポートコントローラUART13の設定後は、割り込みコントローラINTC12の割り込み第1チャンネル、すなわちシリアルポートコントローラUART13からの割り込みを許可状態にする（ステップS902）。その後、通信アプリケーションは待機状態に移行する。

【0093】割り込みが発生すると、通信アプリケーションは割り込みコントローラINTC12を調査する（ステップS903）。そして、割り込み要因を判別し（ステップS904）、割り込み要因がシリアルポートコントローラUART13であれば、割り込み要因をクリアし（ステップS905）、シリアルポートコントローラUART13の受信データを受け取るレジスタからデータを読み出す（ステップS906）。このデータは、座標入力装置PD15に対して入力操作された座標を表している。通信アプリケーションでは、受信した座標に対応する液晶表示装置LCD14の表示用メモリのビットを「1」に設定する（ステップS907）。

【0094】このようにして、通信アプリケーションは、シリアルポートコントローラUART13から受信した座標に対応する液晶表示装置LCD14のピクセルを黒にする。

【0095】通信アプリケーション実行時のシステムエミュレータ20の動作は以下になる。

【0096】インストラクションセットシミュレータISS21に通信アプリケーションの実行を要求すると、インストラクションセットシミュレータISS21は、I/OマネージャIOM27を介して、メモリアクセスエミュレータMAE26から、順次、アプリケーションのプログラムを読み出して命令の実行を行う。

【0097】通信アプリケーションは、最初に、シリアルポートコントローラUART13の通信パラメータの設定を行うために、シリアルポートコントローラUART13の通信パラメータ設定用レジスタにデータを書き込む。

【0098】システムエミュレータ20では、この動作は、インストラクションセットシミュレータISS21が、I/OマネージャIOM27を介して、シリアルポートコントローラエミュレータUARTE23の通信パラメータ設定用レジスタへ、設定データの書き込み要求を送ることによって行われる。シリアルポートコントローラエミュレータUARTE23は、書き込み要求から、通信速度、パリティの有無等の設定を解析し、ウィンドウシステムのアプリケーション・プログラム・インターフェースAPIを使用して、パーソナルコンピュータPCのシリアルポートの設定を行う。書き込み要求には、データ受信時の割り込み許可の設定も含まれてい

23は、割り込み許可設定が指定されていることを判断した場合には、シリアルポートコントローラエミュレータUARTE23の内部に保持されている割り込みの許可状態を、許可に変更する。

【0099】シリアルポートコントローラUART13の設定後、通信アプリケーションは、割り込みコントローラINTC12の割り込み第1チャンネルに割り込み許可の設定を行う。この設定およびシステムエミュレータ20の動作は、描画アプリケーションにおける割り込みコントローラINTC12の設定と同様である。シリアルポートコントローラUART13、割り込みコントローラINTC12の設定が終了すると、通信アプリケーションは待機状態になる。

【0100】この間、シリアルポートコントローラエミュレータUARTE23は、パーソナルコンピュータPCのシリアルポートがデータを受信するのを待っている。この状態では、シリアルポートコントローラエミュレータUARTE23は、一定周期でウインドウシステムのアプリケーション・プログラム・インターフェイスAPIを呼び、パーソナルコンピュータPCのシリアルポートがデータを受信していないかを調べる。受信していれば、シリアルポートコントローラエミュレータUARTE23は、パーソナルコンピュータPCのシリアルポートが受信したデータを、内部の受信データバッファに保持する。

【0101】次に、シリアルポートコントローラエミュレータUARTE23は、データ受信時の割り込み設定を調べ、割り込みが許可されている場合には、割り込みコントローラエミュレータINTCE22に割り込みの発生を通知する。この時、アプリケーションにより、割り込みコントローラINTC12の初期設定が終了していると、シリアルポートコントローラUART13の割り込みは許可に設定されているために、割り込みコントローラエミュレータINTCE22に対して割り込みの発生が通知される。

【0102】これを受けて、割り込みコントローラエミュレータINTCE12は、シリアルポートコントローラエミュレータUARTE23からの割り込みが許可されているかを調べる。許可されている場合には、インストラクションセットシミュレータISS21に対して割り込みの発生を通知する。この時、通信アプリケーションにより割り込みコントローラINTC12の初期設定が終了していると、割り込みが許可されているために、インストラクションセットシミュレータISS21に対して割り込みの発生が通知される。すなわち、通信アプリケーションによる初期設定が終了していれば、データ受信時には、インストラクションセットシミュレータISS21に対して割り込みの発生が通知される。

【0103】割り込みの通知を受けたインストラクションセットシミュレータISS21は、プログラムの実行

を割り込み発生時の処理に切り替える。割り込み発生時に、通信アプリケーションは、割り込みコントローラINTC12を調査して、割り込み要因を調査する。シリアルポートコントローラUART13に割り込み要因がある場合は、アプリケーションは割り込みをクリアした後、シリアルポートコントローラUART13の受信データを取得するレジスタの値を読み出す。

【0104】システムエミュレータ20上にて割り込み要因を調査する動作は、描画アプリケーションの場合と同様である。また、システムエミュレータ20上にて、シリアルポートコントローラUART13からデータを読み出す動作は、イストラクションセットシミュレータISS12がシリアルポートコントローラエミュレータUARTE23の受信データを取り出すレジスタからのデータの読み出しの要求を行い、これを受けたシリアルポートコントローラエミュレータUARTE23が、受信バッファからデータを取り出し、インストラクションセットシミュレータISS21に出力することによって行われる。座標入力装置PD15に対して入力操作された座標が得られると、次に、アプリケーションは、液晶表示装置LCD14に対して点を描画する。点描画時のアプリケーションの動作、システムエミュレータ20の動作は、描画アプリケーションの場合と同様である。

【0105】次に、シリアルポートコントローラエミュレータUARTE23の実現する第2の具体例について説明する。なお、他のペリフェラルエミュレータPEの実現方法は、第1の具体例と同一である。

【0106】パーソナルコンピュータPCには、ターゲットマシン10に搭載されるシリアルポートコントローラと同等のシリアルポートコントローラUART13が搭載されており、パーソナルコンピュータPCから特定のI/Oアドレスにアクセスすることにより、シリアルポートコントローラUART13におけるレジスタの内容にアクセスすることが可能である。

【0107】シリアルポートコントローラエミュレータUARTE23のレジスタに対してアクセスが行われた場合、シリアルポートコントローラエミュレータUARTE23は、アクセスを受けたアドレスからレジスタの種類を判別する。シリアルポートコントローラエミュレータUARTE23は、このレジスタがパーソナルコンピュータPC上のシリアルポートコントローラUART13のどのレジスタに対応するかを判別し、パーソナルコンピュータ10上でこのレジスタが存在するI/Oアドレスに対して、インストラクションセットシミュレータISS21から受けたアクセスと同様のアクセスを行う。

【0108】システムエミュレータ20からパーソナルコンピュータPCのシリアルポートコントローラUART13へのアクセスにより、パーソナルコンピュータPCのシリアルポートの通信速度、ストップビット、バリ

ティ、フロー制御などの各種設定、送信データの書き込み、受信データの読み込み、割り込みの許可禁止の設定などが行える。

【0109】パーソナルコンピュータPCが、パーソナルコンピュータPCのシリアルポートコントローラUART13から割り込みを受けた場合、シリアルポートコントローラエミュレータUARTE23にこれが通知される。これを受けて、シリアルポートコントローラエミュレータUARTE13は割り込みコントローラエミュレータINTCE23に割り込みの発生を通知する。このようにして、シリアルポートコントローラUART13のエミュレーションを行う。

【0110】前述の「通信アプリケーション」を実行させた場合のシステムエミュレータの動作は以下のようになる。

【0111】インストラクションセットシミュレータISS21に通信アプリケーションの実行を要求すると、インストラクションセットシミュレータISS21は、I/OマネージャIOM27を介して、メモリアクセスエミュレータMAE26から、順次、プログラムを読み出して命令を実行していく。

【0112】通信アプリケーションは、最初に、シリアルポートコントローラUART13の通信パラメータの設定を行うために、シリアルポートコントローラUART13の通信パラメータを設定するレジスタに設定データを書き込む。

【0113】システムエミュレータ20では、この動作は、インストラクションセットシミュレータISS21が、I/OマネージャIOM27を介して、シリアルポートコントローラエミュレータUARTE23の通信パラメータを設定するレジスタへ設定データを書き込み、これを受けたシリアルポートコントローラエミュレータUARTE23が受け取った設定データを、パーソナルコンピュータPC上のシリアルポートコントローラUART13の通信パラメータを設定するレジスタへと書きこむことにより行われる。この動作により、パーソナルコンピュータPCのシリアルポートコントローラUART13の通信設定が行われる。

【0114】シリアルポートコントローラUART13の通信設定後、通信アプリケーションは、割り込みコントローラINTC12の割り込み第1チャンネルに割り込み許可の設定を行う。この設定およびシステムエミュレータ20の動作は、第1の具体例での割り込みコントローラINTC12の設定と同様である。シリアルポートコントローラUART13、割り込みコントローラINTC12の設定が終わると、アプリケーションは待機状態になる。

【0115】この間、シリアルポートコントローラエミュレータUARTE23は、パーソナルコンピュータPCのシリアルポートがデータを受信するのを待つため

に、パーソナルコンピュータPCに割り込みが発生するのを待っている。割り込みが発生すれば、割り込みコントローラエミュレータINTCE12に対して割り込みの発生が通知される。これを受けて、割り込みコントローラエミュレータINTCE22は、シリアルポートコントローラエミュレータUARTE23からの割り込みが許可されているかを調べる。許可されている場合には、インストラクションセットシミュレータISS21に対して割り込みの発生を通知する。

【0116】この時、アプリケーションにより割り込みコントローラINTC12の初期設定が終了していると、割り込みが許可されているために、インストラクションセットシミュレータISS21に対して割り込みの発生が通知される。すなわち、通信アプリケーションによる初期設定が終了していれば、データ受信時には、インストラクションセットシミュレータISS22に対して割り込みの発生が通知される。

【0117】割り込みの通知を受けたインストラクションセットシミュレータISS21は、プログラムの実行を、割り込み発生時の処理に切り替える。割り込み発生時に、通信アプリケーションは、割り込みコントローラINTC12を調査して割り込み要因を調査する。シリアルポートコントローラUART13に割り込み要因がある場合は、割り込みをクリアした後、シリアルポートコントローラUART13の受信データを取得するレジスタの値を読み出す。システムエミュレータ20上で割り込み要因を調査する動作は、第1の具体例の場合と同様である。

【0118】システムエミュレータ20上でシリアルポートコントローラUART13からデータを読み出す動作は、インストラクションセットシミュレータISS21がシリアルポートコントローラエミュレータUARTE23の受信データを取り出すレジスタからのデータの読み出しの要求を行い、これを受けたシリアルポートコントローラエミュレータUARTE23がパーソナルコンピュータPCのシリアルポートコントローラUART13の受信データを取得するレジスタからデータを読み出し、インストラクションセットシミュレータISS21へ返すことによって行われる。

【0119】読み込まれたデータは、点を描画すべき座標を表している。座標入力装置PD15に対して入力操作された座標が得られれば、次に、アプリケーションは液晶表示装置LCD14に対して点を描画する。点描画時のアプリケーションの動作、システムエミュレータ20の動作は、第1の具体例における描画アプリケーションの場合と同様である。

【0120】次に、シリアルポートコントローラUART13のエミュレートを実現する第3の具体例について説明する。他のペリフェラルエミュレータPEの実現方法は、第1の具体例と同様である。

【0121】ターゲットマシン10用のアプリケーションの開発環境として、シリアルポートを扱うライブラリが用意される。シリアルポートを扱うアプリケーションは、すべてこのライブラリを介してのみシリアルポートを利用した通信が可能になっている。このライブラリを呼び出すことにより、通信速度、ストップビット、パリティ、フロー制御などの各種通信設定、データ送信、データ受信ができる。ライブラリは、ターゲットマシン10用とシステムエミュレータ20用とが、それぞれ別に用意される。

【0122】ターゲットマシン10用のライブラリのフローチャートを図10に示す。ターゲットマシン10用のライブラリは、シリアルポートの設定が要求されると、設定データを用意し、これをシリアルポートコントローラUART13の設定用レジスタに書き込むことにより通信設定を行う（ステップS1003～S1004）。

【0123】同様に、通信設定を取得する場合は、シリアルポートコントローラUART13の設定用レジスタからデータを読み込むことにより、通信設定を取得する（ステップS1005）。データ送信を行う場合は、シリアルポートコントローラUART13の送信データを設定するレジスタに送信するデータを、1バイト書き込む（ステップS1007）。データ送信には時間がかかるため、データの送信完了を待つ（ステップS1008）。データの送信が完了すると、設定用レジスタの特定のビットが変化する。ライブラリは、これを検知すると、次のデータを、送信データを設定するレジスタに書きこむ。この動作を繰り返すことにより、複数バイトのデータ送信を行う。すべてのデータの送信が完了すると、アプリケーションに処理を戻す。データ受信も同様の方法で行う（ステップS1009～S1010）。

【0124】ペリフェラルエミュレータ用のライブラリのフローチャートを図11に示す。ペリフェラルエミュレータ用のライブラリは、シリアルポートに対する要求を受けた場合、メモリマップ上に配置されたシリアルポートコントローラエミュレータUARTE23の要求通知用レジスタに、要求に対応したデータを書きこむ（ステップS1101）。データが書きこまれると、シリアルポートコントローラエミュレータUARTE23は、書きこまれたデータから要求の内容を解析し（ステップS1102）、ウィンドウシステムのアプリケーション・プログラム・インターフェイスAPIを呼び出し、パーソナルコンピュータPC上のシリアルポートの通信設定およびシリアルポートを用いた通信を行う（ステップS1103）。ウィンドウシステムのシリアルポートに対する処理が終われば、ペリフェラルエミュレータ用のライブラリは、システムエミュレータ20上で動作しているアプリケーションに処理を戻す。

【0125】シリアルポートコントローラエミュレータ

UARTE23が持つレジスタは、ライブラリからシリアルポートコントローラエミュレータUARTE23にシリアルポートに対する要求を渡すためのものであり、実際のハードウェアが持つレジスタとはインターフェースが異なる。このインターフェースは、ペリフェラルエミュレータPEの製作者が、システムエミュレータ20上で動くプログラムからの要求を、効率よくペリフェラルエミュレータPEに通知できるように設計する。

【0126】ペリフェラルエミュレータ用のライブラリは、シリアルポートからデータの受信を行う要求があれば、データの受信を行うという動作の種類、受信したデータを格納するバッファへのアドレス、受信するデータのサイズといったデータを、レジスタを介してシリアルポートコントローラエミュレータUARTE23に渡す。これを受け取ったシリアルポートコントローラエミュレータUARTE23は、パーソナルコンピュータPC上のシリアルポートに対してデータの受信を要求する。レジスタを介して指定されたサイズ分のデータを受信すれば、そのデータをメモリアクセスエミュレータMAE26の受信したデータを格納するバッファへと保存する。データ保存後、ペリフェラルエミュレータPE、ライブラリは処理を完了し、システムエミュレータ20上で動作するプログラムに処理を返す。書き込み動作、通信の設定動作も同様に行われる。

【0127】このような構成とすることにより、他の方式に比べ、ペリフェラルエミュレータPEを大幅に簡略化することができる。

【0128】このシステムエミュレータ20では、通信アプリケーションの内容は多少変わる。アプリケーションは、最初に、シリアルポートコントローラUART13の通信パラメータの設定を行うために、シリアルポートの初期設定を行うライブラリを呼び出す。次に、シリアルポートからデータを読みこむライブラリを呼び出し、データの受信を待つ。データを受信すると、通信アプリケーションに処理が戻るので、アプリケーションは受信したデータをもとに液晶表示装置LCD14に点を描画する。ライブラリがハードウェアに関する処理を行うため、通信アプリケーション自体の処理は簡潔になる。ライブラリ内部の動作は、前述の通りである。

【0129】以上、本実施形態のシステムエミュレータ20の構成及び動作を説明した。ペリフェラルエミュレータPEについては、第1～第3の具体例を示した。簡単なペリフェラルを例として扱ったため、各具体例では、同じペリフェラルが3つの方式で模倣されるが、実際には、エミュレーションの対象となるペリフェラルの機能および複雑度に応じて、システムエミュレーションの模倣方式が3つの具体例のいずれかの方式が選択される。

【0130】

【発明の効果】以上、本発明のシステムエミュレータ

は、インストラクションセットシミュレータとペリフェラルエミュレータとを備えており、インストラクションセットシミュレータをプログラムの実行に用いるために、割り込みなどのターゲットマシンにおける低レベルの動作を模倣できる。ペリフェラルエミュレータは、機能毎に設けられ、それぞれ、模倣する各ペリフェラルの機能、構造の複雑度に応じて模倣方式を変更し、必要に応じて、システムエミュレータ上で行われるターゲットマシン上のペリフェラルへの操作を、実行マシン上のペリフェラルへの操作に変換することができるため、外部機器との通信等、従来のシステムエミュレータでは行えなかった動作が可能になる。

【0131】これにより、本発明のシステムエミュレータは、ターゲットマシンと同様の実行環境を提供することができ、プログラムの開発効率を向上させることができる。また、ペリフェラルの機能、複雑度に応じて、模倣方式を変更することにより、ペリフェラルエミュレータ自体の開発効率及び信頼性も向上させることができる。

【0132】また、シリアルポートコントローラエミュレータが、パーソナルコンピュータのシリアルポートに他の機器を接続した状態で動作を実行できる構成とすることによって、例えば、ターミナル等をシリアルポートに接続して、システムエミュレータ上で動いているプログラムと通信を行うことができる。

【0133】また、液晶表示装置エミュレータがターゲットマシンの表示をパーソナルコンピュータ上でを行い、座標入力装置エミュレータが表示の上でのマウスの操作を座標の入力手段として用いることによって、液晶表示装置に表示された画面をペンで押すという動作を、パーソナルコンピュータのディスプレイに表示された画面をマウスでクリックするという動作で実現できる。

【0134】このように、本発明のシステムエミュレータでは、ターゲットマシンでの操作が、そのままシステムエミュレータ上で実施されるために、テスト、デバックが可能になる。

【0135】以上のように、本発明のシステムエミュレータを用いることにより、実行マシン上でターゲットマシンと同等な環境でのプログラムの実行が行えるため、コスト、時間的に負担の大きいインサーキットエミュレータを用いることなく、周辺装置を含めた環境でのデバックが行え、プログラムの開発効率を向上させることができる。

【図面の簡単な説明】

【図1】本発明のシステムエミュレータの実施形態にお

いて、模倣対象とされるターゲットマシンの構成図である。

【図2】そのターゲットマシンのメモリマップである。

【図3】本発明のシステムエミュレータの実施の形態の一例を示す構成図である。

【図4】そのシステムエミュレータにおけるインストラクションセットシミュレータの構成図である。

【図5】そのシステムエミュレータに使用されるI/Oマネージャの動作説明のためのフローチャートである。

【図6】そのシステムエミュレータに使用される割り込みコントローラエミュレータの動作説明のためのフローチャートである。

【図7】そのシステムエミュレータに使用されるシリアルポートコントローラエミュレータの動作説明のためのフローチャートである。

【図8】そのシステムエミュレータに使用される描画アプリケーションの動作説明のためのフローチャートである。

【図9】そのシステムエミュレータに使用される通信アプリケーションの動作説明のためのフローチャートである。

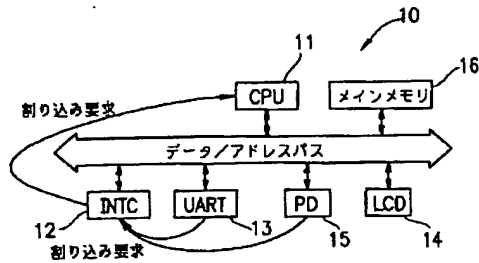
【図10】ターゲットマシン用シリアルポートライブラリの動作説明のためのフローチャートである。

【図11】実行マシン用のシリアルポートライブラリの動作説明のためのフローチャートである。

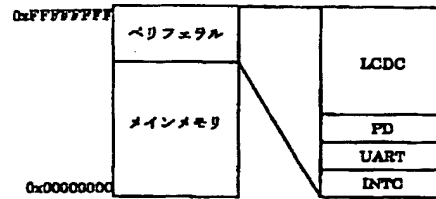
【符号の説明】

- 10 ターゲットマシン
- 11 CPU
- 12 割り込みコントローラINTC
- 13 シリアルポートコントローラUART
- 14 液晶表示装置LCD
- 15 座標入力装置PD
- 20 システムエミュレータ
- 21 インストラクションセットシミュレータISS
- 22 割り込みコントローラエミュレータINTCE
- 23 シリアルポートコントローラエミュレータUARTE
- 25 座標入力装置エミュレータPDE
- 24 液晶表示装置エミュレータLCDE
- 26 メモリアクセスエミュレータMAE
- 27 I/OマネージャIOM
- GR0~GR3 汎用レジスタ
- PrC プログラムカウンタ
- FR フラグレジスタ

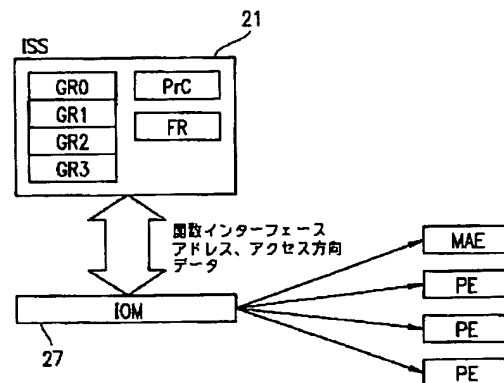
【図1】



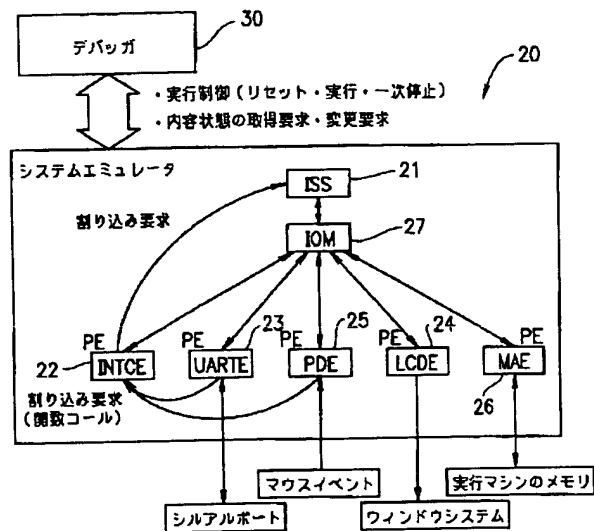
【図2】



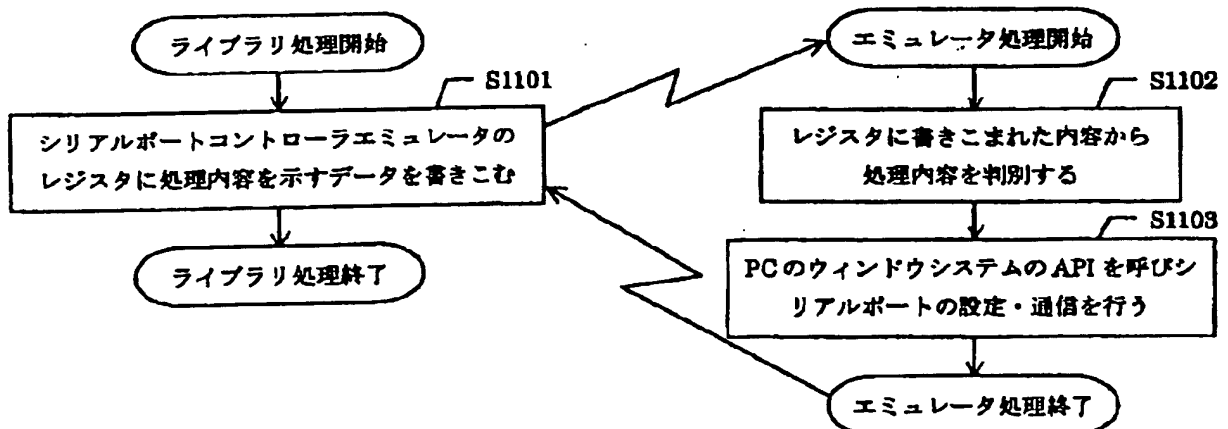
【図4】



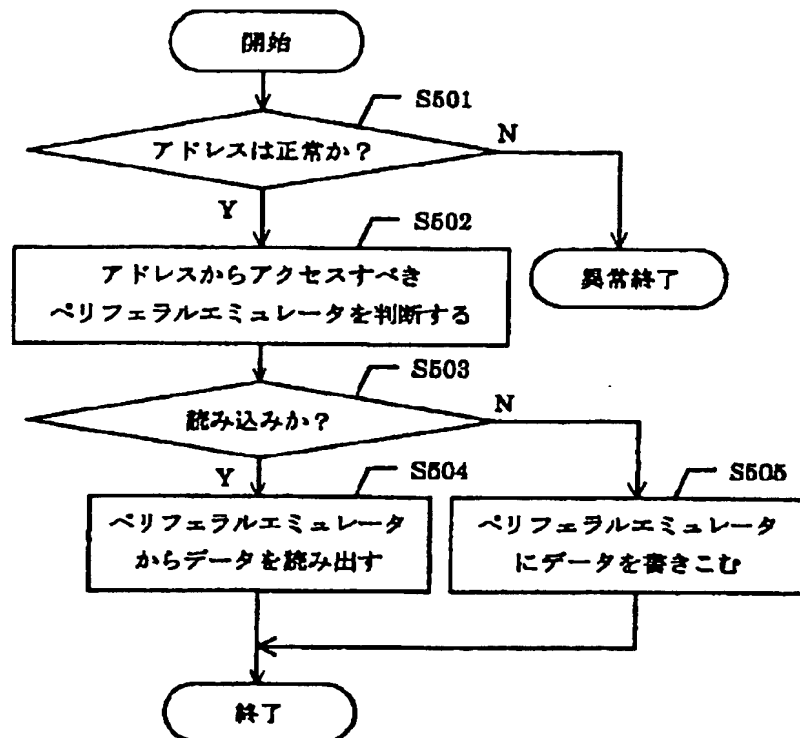
【図3】



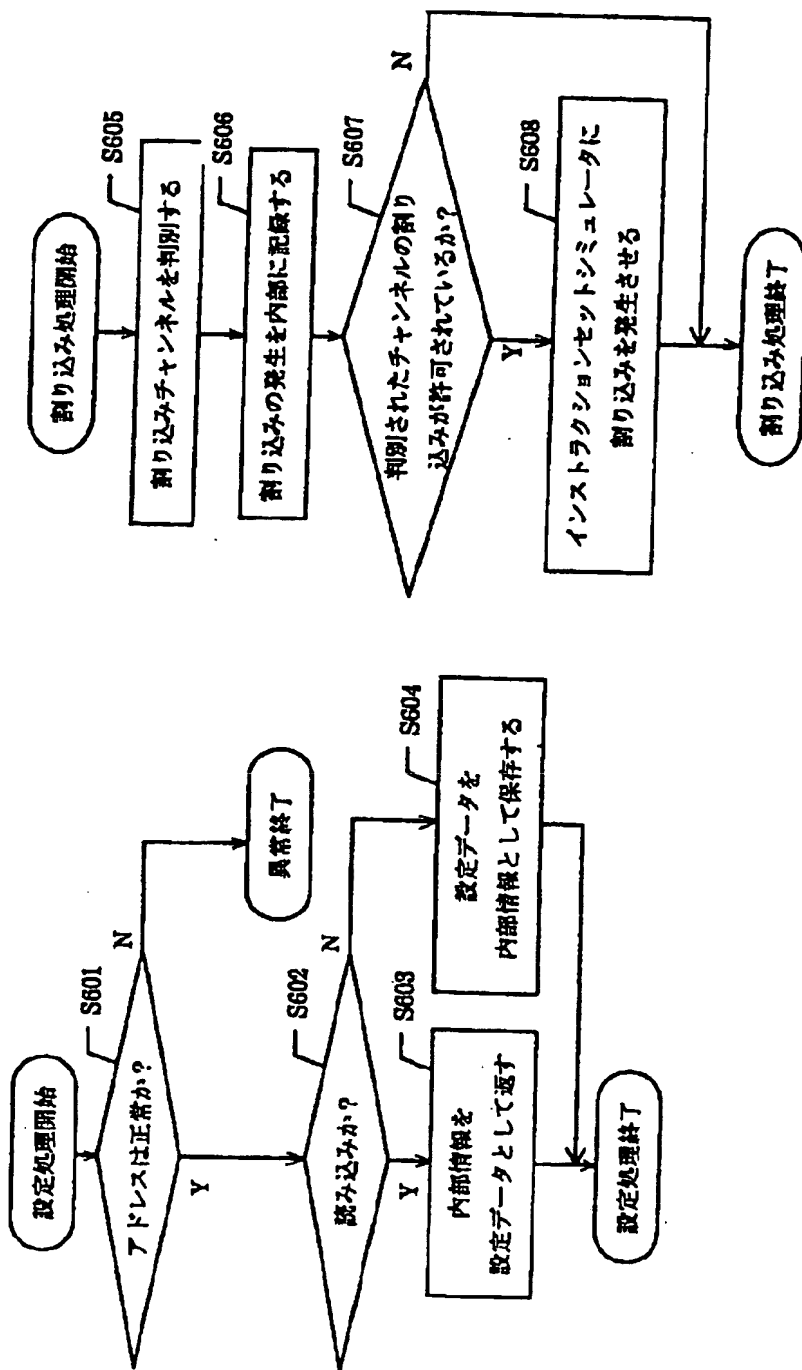
【図11】



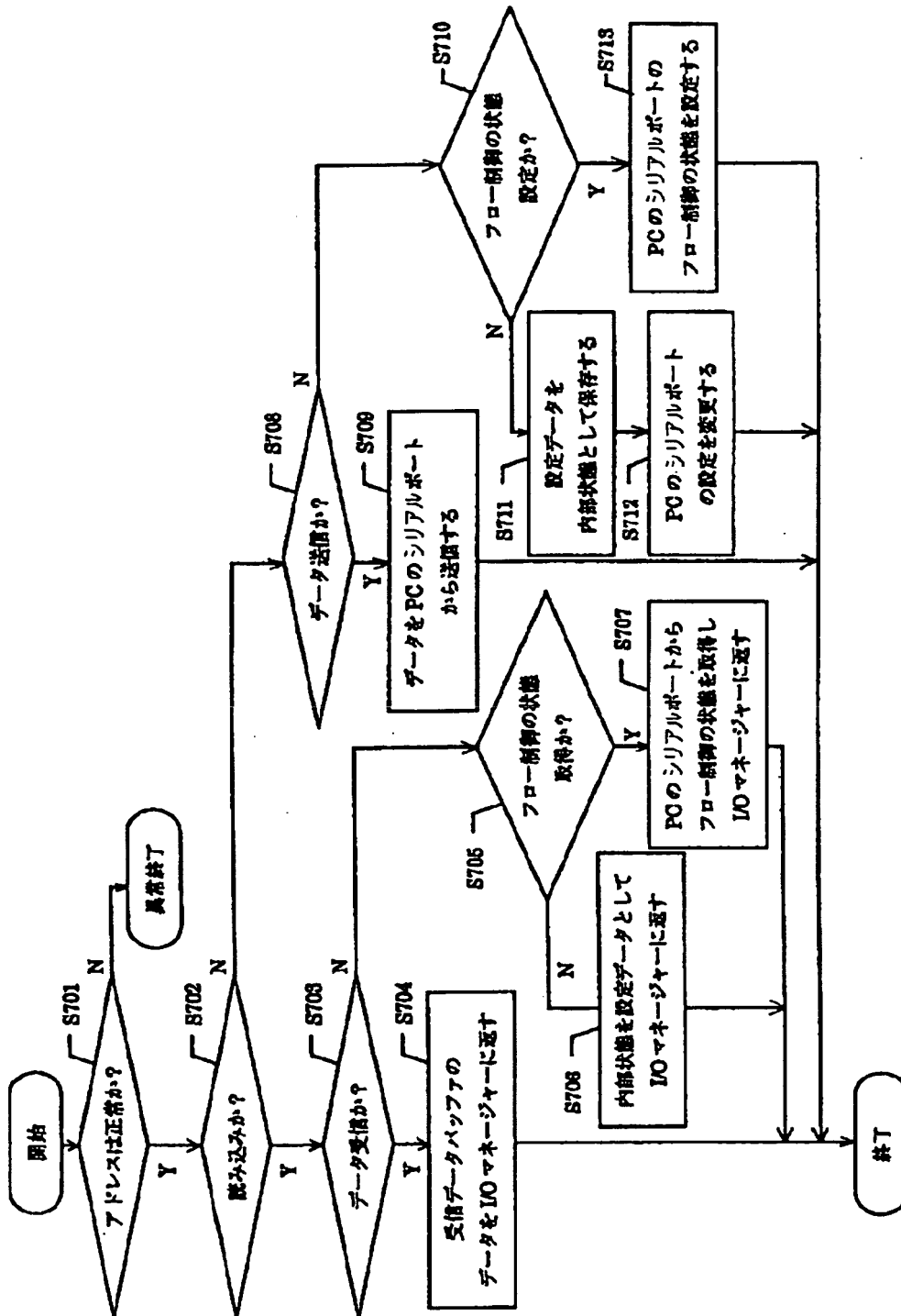
【図5】



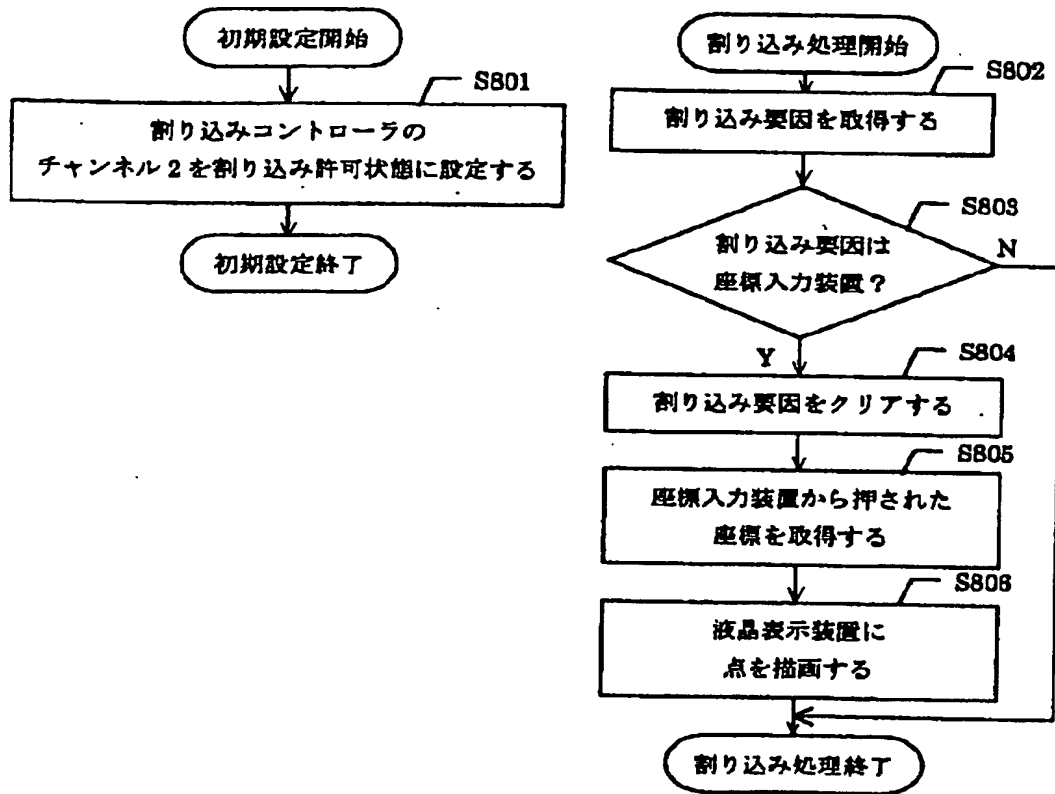
【図6】



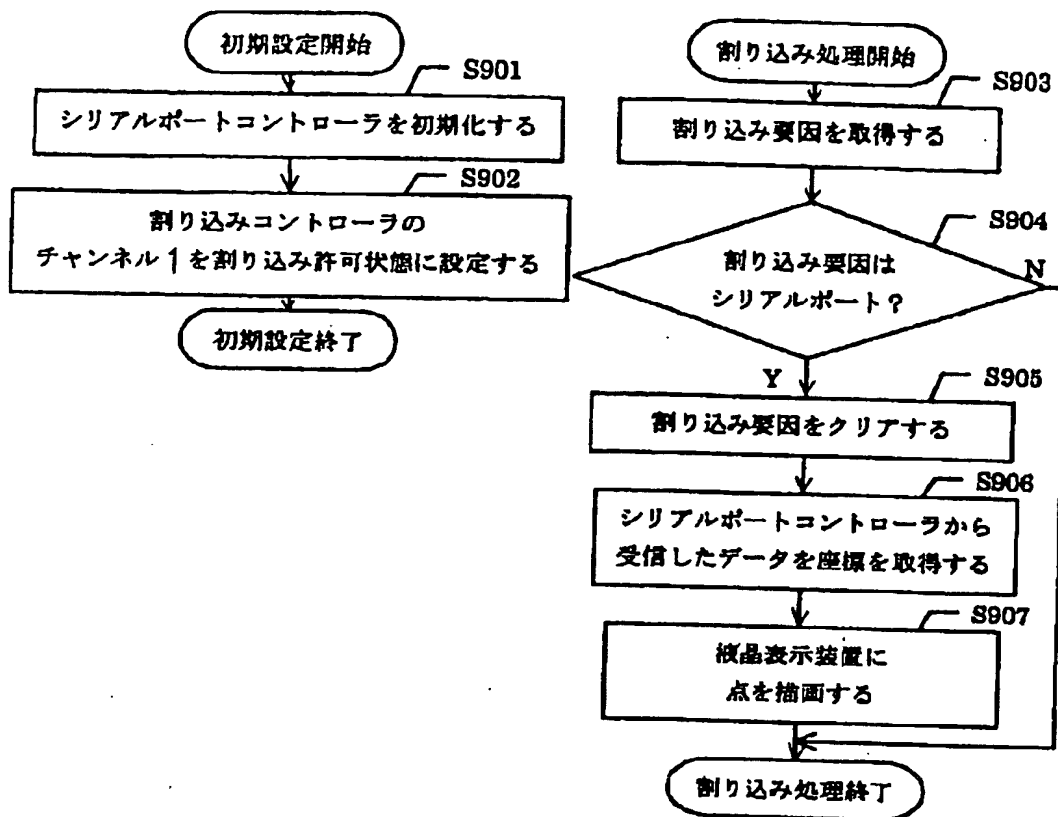
【圖 7】



【図8】



【図 9】



【図10】

